

Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування

Кафедра будівельних, дорожніх, меліоративних,  
сільськогосподарських машин і обладнання



**02-01-511М**

## **МЕТОДИЧНІ ВКАЗІВКИ**

до виконання практичних робіт  
з навчальної дисципліни **«Робототехніка в машинобудуванні»** для  
здобувачів вищої освіти першого (бакалаврського) за освітньо-  
професійною програмою «Галузеве машинобудування»  
спеціальності 133 «Галузеве машинобудування» всіх форм навчання

Рекомендовано навчально-  
методичною радою з якості  
навчально-наукового  
механічного інституту,  
Протокол № 7 від 23.02.2021 р.

Рівне – 2021

Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Робототехніка в машинобудуванні» для здобувачів вищої освіти першого (бакалаврського) за освітньо-професійною програмою «Галузеве машинобудування» спеціальності 133 «Галузеве машинобудування» всіх форм навчання [Електронне видання] / Голотюк М. В., Реут Д. Т., Марчук Н. М. – Рівне : НУВГП, 2021. – 35 с.

Укладачі:

Голотюк М. В., к.т.н., доцент кафедри будівельних, дорожніх, меліоративних, сільськогосподарських машин і обладнання;

Реут Д. Т., старший викладач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій;

Марчук Н. М., к.т.н., доцент кафедри автомобілів та автомобільного господарства.

Відповідальний за випуск: Кравець С. В., д.т.н, професор, завідувач кафедри будівельних, дорожніх, меліоративних, сільськогосподарських машин і обладнання.

Керівник групи забезпечення спеціальності 133 «Галузеве машинобудування»: С.В. Кравець, д.т.н, професор, завідувач кафедри будівельних, дорожніх, меліоративних, сільськогосподарських машин і обладнання

© М. В. Голотюк, Д. Т. Реут,  
Н. М. Марчук, 2021  
© НУВГП, 2021

## ЗМІСТ

Загальні положення.....	4
Методичні рекомендації до виконання практичних занять.....	5
<b>Практична робота 5.</b> Розробка моделі робота-маніпулятора ..	5
<b>Практична робота 6.</b> Основи програмування в середовищі Arduino IDE. Вивчення роботи з вхідними та вихідними дискретними сигналами .....	15
<b>Практична робота 7.</b> Проектування організації зчитування сигналів з датчиків .....	23
<b>Практична робота 8.</b> Дослідження роботи сервоприводів та реалізація циклограми .....	29
Рекомендована література.....	35

## ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дисципліна «Робототехніка в машинобудуванні» є невід’ємним складником формування професійної компетентності студентів. Програма дисципліни передбачає комплексне вивчення робототехніки в машинобудуванні.

Робототехніка в машинобудуванні відносно самостійна дисципліна, яка дає загальне уявлення про розвиток техніки. Даний курс покликаний сприяти формуванню у студентів технічних спеціальностей загальної картини розвитку інженерної справи як цілісного процесу, який відбувається закономірно і проходить в органічному взаємозв’язку і взаємодії з історією суспільства.

Курс «Робототехніка в машинобудуванні» дозволить отримати знання в обсязі, достатньому для самостійного вирішення конструкторських та виробничо-технологічних завдань в галузі конструювання, проектування та сервісного обслуговування робототехнічних систем та комплексів, призначених для автоматизації виробничих (технологічних) процесів. Отримані вміння дозволяють використовувати інженерні методики, аналітичні та числові методи розрахунку для аналізу відомих та розробки нових механізмів, вузлів та комплексів обладнання робототехнічних систем.

**Мета** навчальної дисципліни «Робототехніка в машинобудуванні» є надання студентам знань і навиків в області розвитку та використання роботів і роботизованих технологічних комплексів (РТК), ознайомлення з методикою вибору та проектування РТК для вирішення технологічних задач виробництва..

У результаті вивчення навчальної дисципліни «Робототехніка в машинобудуванні» студент повинен *знати*: типові та сучасні конструкції промислових роботів, принципи їх функціонування та сфери використання, переваги і недоліки; інженерні методики розрахунку та особливості проектування промислових роботів, методи розрахунку їх основних вузлів і механізмів на міцність, жорсткість, стійкість та довговічність; методи кінематичних і динамічних розрахунків, комп’ютерного моделювання та аналізу механізмів робота.

# МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ПРАКТИЧНИХ ЗАНЯТЬ

## Практична робота 5

### Тема: Розробка моделі робота-маніпулятора

**Мета роботи:** Вивчити методику розробки моделі робота-маніпулятора

#### Теоретичні відомості

Маніпулятор робота за своїм функціональним призначенням повинен забезпечувати рух вихідної ланки і, закріпленого в ньому, об'єкта маніпулювання в просторі по заданій траєкторії і з заданою орієнтацією. Для повного виконання цієї вимоги основний механізм важеля маніпулятора повинен мати не менше шести степеней вільності, причому рух по кожній з них має бути керованим. Робот з шістьма степенями вільності є складною автоматичною системою. Ця система складна як у виготовленні, так і в експлуатації. Тому в реальних конструкціях роботів часто використовуються механізми з числом подвижностей менше шести. Найбільш прості маніпулятори мають три, рідше дві, степені вільності. Такі маніпулятори значно дешевше у виготовленні та експлуатації, але висувають специфічні вимоги до організації робочого середовища.

Модель маніпулятора складена на основі аналізу існуючих конструкцій побутових роботів. Основним режимом роботи маніпулятора є безперервне стеження за об'єктом.

Таким чином, завдання данної роботи полягає в аналітичному рішенні прямої та зворотної задачі кінематики у вигляді, придатному для алгоритмічного опису та написання керуючої програми, виконання якої відбувається в режимі реального часу, вкладається в проміжки між отриманими даних про об'єкт від відеокамери.

На основі проведеного аналітичного огляду сформовано технічне завдання на конструювання робота-маніпулятора для позиціонування датчиків на виробах зі складною геометрією (рис.5.1.).

Для реалізації задуманої конструкції робота-маніпулятора проаналізовано роботи-маніпулятори промислові, та зпроектувати

прототип промислового робота-маніпулятора. Даний робот-маніпулятор розроблений для проведення певних операцій, які автоматизують промислові процеси. Також можливе використання даного робота у процесі контролю, при оснащенні активної частини робота-маніпулятора відповідними датчиками, наприклад ультразвуковими.



Рис. 5.1. Модель робота-маніпулятора

Для зручності реалізації поставленої задачі було вирішено розділити конструкцію робота-маніпулятора на три частини:

- активна частина робота-маніпулятора;
- рукоятка;
- робоча зона або зона захвату.

Розділення на частини даної моделі обумовлено характеристиками 3D-принтера на якому було заплановано друк деталей, а саме: розмірні характеристики робочої поверхні принтера.

До активної частини робота-маніпулятора входить стаціонарна зона, у якій розміщується серводвигун та підшипники. Разом ці деталі забезпечують переміщення рукоятки разом з робочою частиною навколо своєї осі. Окрім цього, активна частина відповідає

за положення рукоятки у просторі. На рисунку 5.2 показано внутрішні складові активної частини.



Рис. 5.2. Внутрішні складові активної частини робота-маніпулятора

Для реалізації створення активної частини робота-маніпулятора створити її модель в програмному середовищі SolidWorks, та розділити на складові частини, які згодом роздруковуються на 3D-принтері. На рисунку 5.3 зображено деталі – дискові основи активної частини, які скріплюються між собою гвинтами та містять всередині підшипник, який згодом буде частиною обертального руху активної частини.



Рис. 5.3. Складові активної частини після друку на 3D-принтері

Для повного функціонування на зборі активної частини моделі робота- маніпулятора необхідні такі матеріали:

- Кроковий двигун Nema 17;
- Гвинти М3 х 10мм та М3 х 8мм;
- Шурупи М3 х 3 мм та М4 х 3 мм;
- Підшипник 10 мм х 30 мм х 9 мм, 6200zz
- Сталеві кульки 10 мм.

Для створення 3D-моделей було застосувати програмне середовище SolidWorks. А для адаптування моделі під параметри 3-Д принтера, та задання розмірів, матеріалу та щільності друку застосувати програмний продукт Ultimaker Cura.

На рисунку 5.4 зображено змодельований вигляд активної частини робота- маніпулятора, в якому всередині встановлено кроковий двигун, підшипник та 10 сталевих кульок.

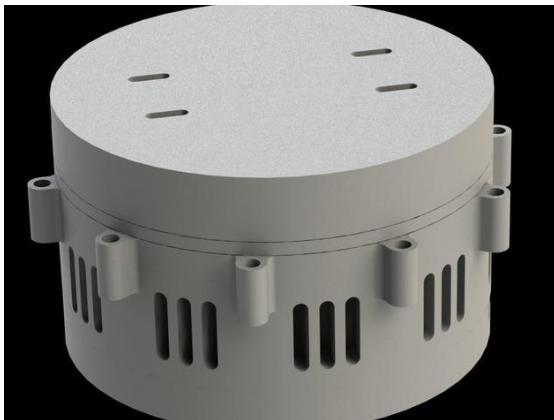


Рис. 5.4. Змодельована активна частина робота-маніпулятора

Сталеві кульки призначені для обертання верхньої частини конструкції по своїй осі, таким чином забезпечуватиметься рух рукоятки на 360°. Також верхня та нижня частина конструкції з'єднуються між собою по колу гвинтами, в сумарній кількості присутні 10 з'єднань по колу (рис. 5.5).



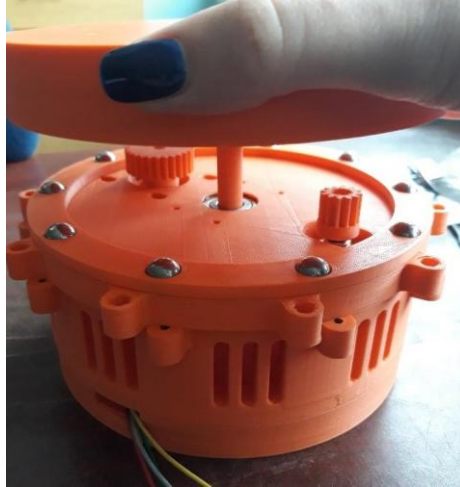


Рис. 5.5. Роздрукована на 3D-принтері та зібрана активна частина робота-маніпулятора

При друку на 3D-принтері деталей застосовано пластик PLA, з такими механічними властивостями:

- міцність на розтяг 51 МПа;
- відносне подовження при розриві 30%;
- модуль пружності при розтягу 2300 МПа;
- модуль пружності при згинанні 1440;
- міцність при згинанні 80 Мпа.

Даний матеріал було обрано з розрахунку міцності подальшої конструкції. За своїми властивостями пластик типу PLA має більш міцну характеристику ніж пластик типу ABS.

Кінцевий вигляд 1/3 конструкції робота-маніпулятора, а саме активної частини робота-маніпулятора зображено на рисунку 5.6.

Всередині активної частини розміщено такі елементи активного переміщення як: кроковий двигун Nema17 та потенціометр BOURNS 3590S.

Для роботи рукоятки в парі з захватом та активною частиною встановлено 5 серводвигунів типу MG995. Даний двигун було підібрано на основі проаналізованих прикладів роботів-

маніпуляторів, а також у задоволенні характеристиками двигуна вимог, які було висунуто до даної частини конструкції.

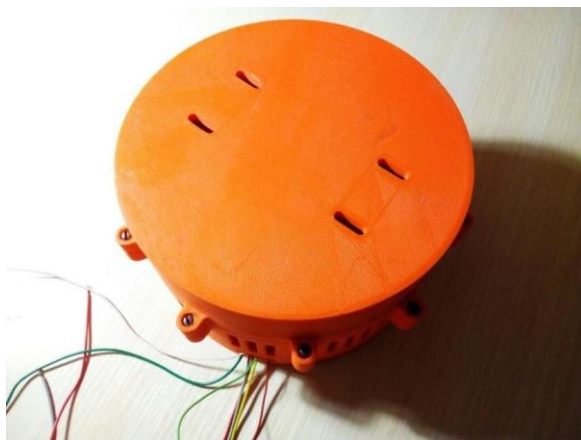


Рис. 5.6. Активна частина робота-маніпулятора в зібраному стані

Основною задачею рукоятки та її плечей є передача крутного моменту від активної частини до захвату, таким чином робот-маніпулятор матиме 5 ступенів свободи. Для реалізації складання даної конструкції необхідно було використати такі матеріали:

- Гвинти М3 х 8мм, М3 х 10мм, М3 х 63мм та М3 х 16мм;
- Підшипники типу 688zz з розміром 8 мм х 16 мм х 5 мм;
- Гайка М3 10 мм;
- Серводвигун MG995.

Модель надруковано на 3-Д принтері деталі, як зображено на рисунку 5.7.

Змодельований вигляд рукоятки та плечей представлено на рисунку 30.

Всі отвори було зашліфовано так як при друку неможливо досягнути точного розміру отвору під гвинт та підшипники. При друку деталей рукоятки щільність заповнення деталей матеріалом була виставлена не менше ніж 50%.

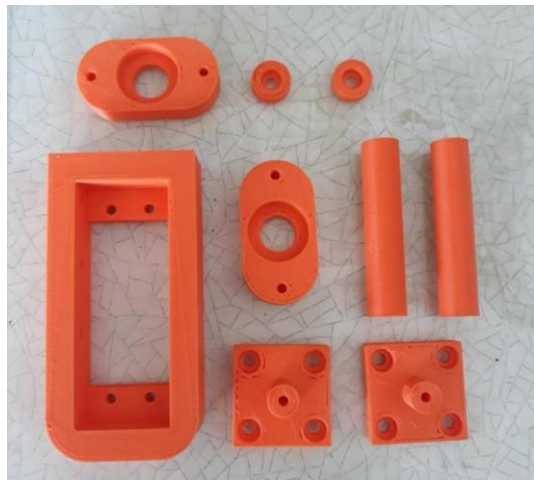


Рис. 5.7. Надруковані деталі рукоятки робота-маніпулятора



Рис. 5.8. Модель рукоятки та плечей

Кінцевий вигляд іншої частини конструкції робота-маніпулятора, а саме рукоятки робота-маніпулятора зображено на рисунку 5.9.



Рис. 5.9. Рукоятка робота-маніпулятора в зібраному стані

Як видно з рисунку 5.9 встановлено 3 серводвигуна, які будуть забезпечувати переміщення рукоятки.

Проте, однією з найважливіших елементів робота-маніпулятора є зоназахвату. Для реалізації роботи зони захвату було застосовано такі матеріали:

- Гвинти М3 x 25мм, М3 x 10 мм, М3 x 8мм;
- Підшипники типу 688zz
- Серводвигун MG995.

Розрдуковану зону захвату зображено на рисунку 5.10. Всі отвори зашліфовано так як при друку неможливо досягнути точного розміру отвору під гвинт та підшипники. При друку деталей рукоятки щільність заповнення деталей матеріалом була виставлена не менше ніж 70%.

З'єднання зони захвату виконане у вигляді двох деталей з перфорованою внутрішньою частиною, що необхідне для чіткого захвату деталей при роботі з роботом-маніпулятором.

Дана частина робота-маніпулятора з'єднується з рукояткою завдяки окремому елементу, який є з'єднувальною частиною між зоною захвату та рукояткою (рис.5.11). Сама зона захвату з'єднується за допомогою гвинтів.



Рис. 5.10. Зона захвату робота-маніпулятора в зібраному стані



Рис. 5.11. Зона захвату з кріпильними елементами в зібраному стані

Після друку на 3D-принтері всіх деталей, їх шліфування та збору отримано завершену модель робота-маніпулятора для позиціонування датчиків на виробах зі складною геометрією, яку зображено на рисунку 5.12.



Рис. 5.12. Модель робота-маніпулятора в зібраному стані

Завершена модель зібраного робота-маніпулятора готова до програмування та подальшого використання згідно запланованих цілей застосування.

В роботі описано процес створення конструкції моделі робота-маніпулятора. Обрано комплектуючі елементи та надано опис їх застосування у моделі. Також за допомогою системи автоматичного проектування SolidWorks розроблено 3D модель приладу, яку згодом було використано як основу для створення макету. Деталі надруковано на 3D-принтері та зібрано згідно запланованій конструкції. Показано основні вузли робота-маніпулятора та описано принцип їх з'єднання та взаємодії. Конструкція зібрана та готова до програмування.

## Практична робота 6.

### **Тема: Основи програмування в середовищі Arduino IDE. Вивчення роботи з вхідними та вихідними дискретними сигналами**

**Мета роботи:** Ознайомитися з апаратною та програмною частинами середовища Arduino. Вивчити основи роботи з аналоговими і дискретними сигналами в Arduino IDE.

#### **Теоретичні відомості**

Arduino – це відкрита електронна платформа, що базується на простому у використанні апаратному та програмному забезпеченні.

Апаратна частина включає мікроконтролерну плату, а програмна – середовище Arduino IDE. Платформа Arduino є відкритою, тому існує ряд аналогів Arduino-сумісних мікроконтролерних плат різного виконання.

#### ***Arduino Uno***

**Arduino Uno** – це пристрій на основі мікроконтролера ATmega328P. До його складу входить все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів/виходів (з них 6 можуть використовуватися в якості ШІМ-виходів), 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньосхемного програмування (ICSP) і кнопка скидання. Для початку роботи з пристроєм достатньо просто подати живлення від AC/DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю.

#### ***Входи і виходи***

Кожен з 14 цифрових виводів може працювати в якості входу або виходу. Рівень напруги на выводах обмежений 5В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Усі виводи з'єднані з внутрішніми підтягуючими резисторами (за замовчуванням відключеними) номіналом 20-50 кОм. Крім цього, деякі виводи Arduino Uno можуть виконувати додаткові функції:

**Послідовний інтерфейс:** виводи 0 (RX) і 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці виводи з'єднані з відповідними

выводами мікросхеми ATmega8U2, що виконує роль перетворювача USB/UART.

**Зовнішні переривання:** виводи 2 і 3. Можуть служити джерелами переривань, що виникають при фронті, спаді або при низькому рівні сигналу на цих висновках.

**ШИМ:** виводи 3, 5, 6, 9, 10 і 11. За допомогою функції analogWrite() можуть виводити 8-бітові аналогові значення у вигляді ШИМ-сигналу.

**Інтерфейс SPI:** виводи 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Із застосуванням бібліотеки SPI дані виводи можуть здійснювати зв'язок по інтерфейсу SPI.

**Світлодіод:** 13. Вбудований світлодіод, приєднаний до виводу 13.

В Arduino Uno є 6 аналогових входів (A0 - A5), на кожен з яких можна подати напругу в межах 0 – 5В та отримати аналогово-цифрове перетворення вхідного сигналу у вигляді 10-бітного числа (1024 різних значення). Верхню межу вхідної напруги можна змінити, використовуючи вивід AREF і функцію analogReference().

Таблиця 6.1

**Основні характеристики Arduino Uno**

Мікроконтролер	ATmega328
Робоча напруга	5V
Вхідна напруга (рекомендована)	7-12V
Дискретні (цифрові) входи/виходи	14 (6 із них можуть використовуватися як ШИМ-виходи)
Аналогові входи	6
Максимальний струм одного вивода	40 mA
Максимальний струм вивода 3,3В	50 mA
Флеш-пам'ять	32 KB (ATmega328) з яких 0.5 KB використовуються завантажувачем
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Тактова частота	16 MHz



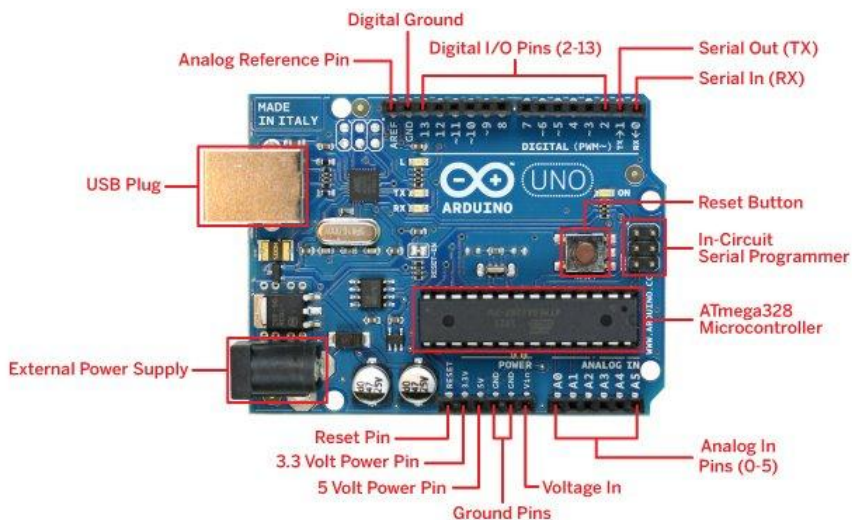


Рис. 6.1.Зовнішній вигляд та призначення виводів плати Arduino Uno

### *Arduino Mega 2560*

Arduino Mega 2560 – це пристрій на основі мікроконтролера ATmega2560.

Основні характеристики плати Arduino Mega 2560 наведені в табл. 5.2.

Таблиця 6.2

### Основні характеристики Arduino Mega 2560

Мікроконтролер	ATmega2560
Робоча напруга	5V
Вхідна напруга (рекомендована)	7-12V
Дискретні (цифрові) входи/виходи	54 (15 із них можуть використовуватися як ШІМ-виходи)

Аналогові входи	16
Максимальний струм одного вивода	40 mA
Максимальний струм вивода 3,3В	50 mA
Флеш-пам'ять	256 KB з яких 8 KB використовуються авантажувачем
SRAM	8 KB
EEPROM	4 KB
Тактова частота	16 MHz

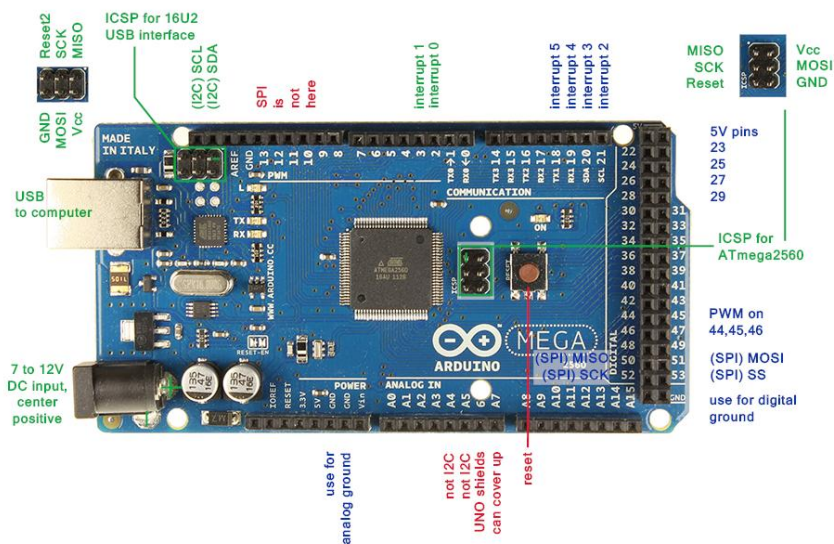


Рис. 6.2. Зовнішній вигляд та призначення виводів плати Arduino Mega 2560

### *Програмування Arduino*

Програмування мікроконтролерних плат Arduino здійснюється у вільно розповсюджуваному середовищі Arduino IDE.

Мікроконтролери в платах Arduino випускаються з прошитим завантажувачем (bootloader), що дозволяє завантажувати в мікроконтролер нові програми без необхідності використання зовнішнього програматора. Взаємодія з ним здійснюється за оригінальним протоколом STK500.

Також мікроконтролер можна прошити і через роз'єм для внутрішньосхемного програмування ICSP (In-Circuit Serial Programming), не звертаючи уваги на завантажувач.

Мова програмування Arduino – це видозмінена мова C/C++ із розширеним набором функцій.

Кожна програма для пристроїв Arduino містить дві основні функції:

1) void setup() – функція, що виконується о дин раз, після кожної подачі живлення або скидання плати Arduino;

2) void loop() – виконується постійно в циклі після виконання функції void setup().

Перед завантаженням програми в мікроконтролер слід спочатку вибрати тип плати Arduino та порт, до якого вона підключена.

### ***Деякі функції для роботи з входами/виходами Arduino***

pinMode(pin, value) – налаштування виводу з номером pin як дискретного входу (value=INPUT) або дискретного виходу (value=OUTPUT). Наприклад:

```
pinMode(13, OUTPUT);
```

digitalRead(pin) – функція зчитує із заданого входу значення HIGH або LOW.

digitalWrite(pin, value) – подає на цифровий вхід/вихід значення HIGH або LOW;

analogRead(pin) – зчитує величину напруги з аналогового входу і видає результат у вигляді цілого числа від 0 до 1023.

### **План роботи**

1. Ознайомитися з будовою та призначенням елементів плати Arduino Uno.

2. Ознайомитися з принципами програмування в середовищі Arduino IDE.

3. Навчитися підключати давачі з дискретними та аналоговими вихідними сигналами до Arduino та обробляти дані цих давачів.

### **Порядок виконання роботи**

1. Ознайомитися з теоретичними відомостями.
2. Завантажити середовище Arduino IDE.
3. Ознайомитись зі схемою підключення механічної кнопки (рис. 6.3.). Зібрати її на макетній платі. Використати підтягуючий резистор на 7,5 кОм.

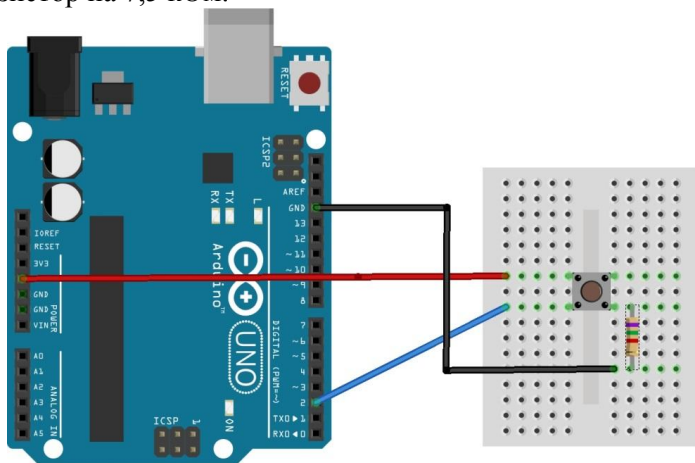


Рис. 6.3. Підключення кнопки до Arduino Uno

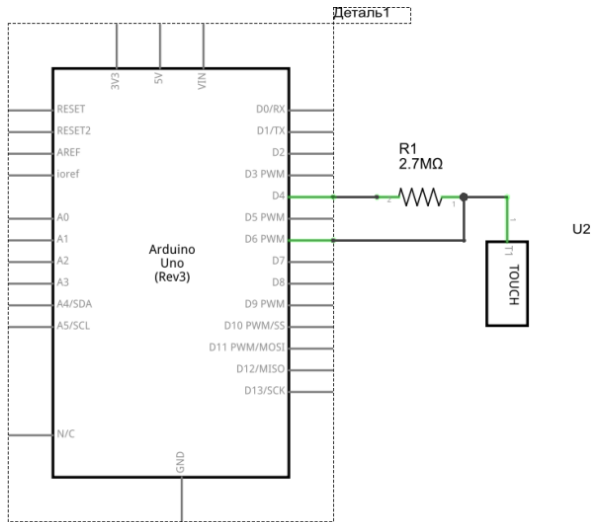
У середовищі Arduino IDE написати програму, що запалюватиме світлодіод при кожному натисканні кнопки.

Скомпілювати програму та переконатися у відсутності помилок.

Підключити мікроконтролерну плату до USB-порта комп'ютера.

У середовищі Arduino IDE вибрати тип плати та порт, до якого вона підключена. Завантажити створену програму в мікроконтролер. Переконатися в коректності роботи програми.

4. Ознайомитися зі схемою підключення ємнісної сенсорної кнопки. Зібрати її на макетній платі. Підключити сенсорну кнопку до 4 (send pin) і 6 (receive pin) дискретних виводів Arduino. Використати резистор 2,5 МОм.



Завантажити бібліотеку Capacitive Sensing

<http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.Cap>

Sense та інсталиувати її. Написати програму, що запалюватиме світлодіод при кожному натисканні кнопки.

```
#include <CapacitiveSensor.h>
```

```
CapacitiveSensor key1 = CapacitiveSensor(4, 6); //
```

підключена сенсорна кнопка між 4-м і 6-м пінами

```
int range = 100; // порогове значення, що визначає
```

натискання кнопки. Підбирається експериментально

```
byte button; // побітове позначення натиснутих
```

кнопок

```
void setup()
```

```
{
```

```
pinMode(13, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
long total = key1.capacitiveSensor(30);  
button = 0;  
    if (total > range) {button=1;}  
if (button==1) digitalWrite(13, HIGH);  
    else digitalWrite(13, LOW);  
}
```

Завантажити програму в мікроконтролер.

Поекспериментувати з чутливістю спрацювання кнопки.

5. Зробити висновки. Звіт повинен містити: титульний лист; тему, мету роботи; порядок виконання; створені програми; висновки.

## Практичне заняття 7

### Тема: Проектування організації зчитування сигналів з давачів

**Мета роботи:** Навчитися програмувати плату Arduino Uno для зчитування сигналу з давачів.

#### Теоретичні відомості

Мікроконтролер в Arduino Uno містить 3 таймери. Вони можуть використовуватись для отримання імпульсного сигналу від давачів.

Мікроконтролер ATmega328P містить два 8-розрядні таймери та один 16-розрядний. Таймер 1 має регістр захоплення OCR1, куди записується вміст таймера TCNT1 по події захоплення - зміні сигналу на вході ICP1 або виході аналогового компаратора.

**Режим захоплення (capture).** У режимі захвату 16-ти бітне значення таймера (TCNT1) захоплюється в регістр OCR1 при кожній події на вході ICP1 або виході аналогового компаратора. Подія для захоплення задається в регістрі TCCR1B (біт ICES1) та ACSR (біт ACIC).

Режим захоплення використовується для вимірювання тривалості між двома подіями, наприклад періоду, тривалості імпульсу, скважності і т.д.

*Приклад 1. Вимірювання періоду дискретного сигналу (рис. 7.1)*

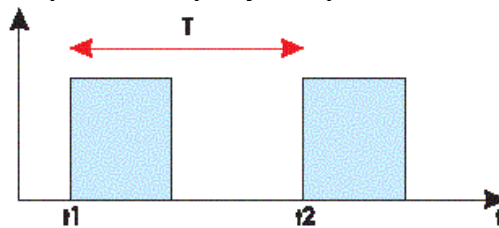


Рис. 7.1. Вимірювання періоду

*Приклад 2. Вимірювання періоду з усередненням результату (рис. 7.2)*

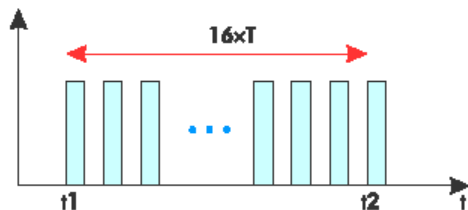


Рис. 7.2. Вимірювання періоду з усередненням

*Приклад 3. Вимірювання тривалості імпульсу (рис. 7.4)*

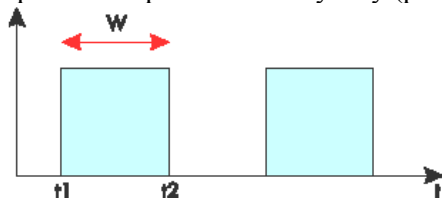


Рис. 7.3. Вимірювання тривалості імпульсу

*Приклад 4. Вимірювання скважності імпульсів (рис. 7.4)*

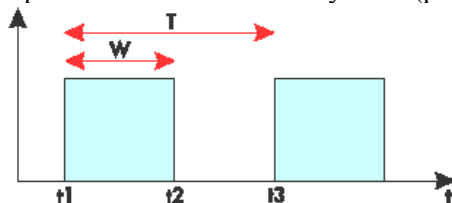


Рис. 7.4. Вимірювання скважності імпульсів

**Аналогово-цифровий перетворювач** (АЦП, англ. Analog-to-digital converter, ADC) - пристрій, що перетворює вхідний аналоговий сигнал в цифровий код (цифровий сигнал).

У мікроконтролері ATmega328P є 10-розрядний АЦП. Вхідна напруга конвертується в 10-розрядне двійкове значення. Мінімальне значення відповідає 0, а максимальне - опорній напрузі. Отриманий результат перетворення записується в 2 регістра: ADCH і ADCL та повертається функцією `analogRead(pin)` як число в діапазоні 0...1023. Результат перетворення можна розрахувати за формулою



$$ADC = \frac{V_{in} \cdot 1024}{V_{ref}}$$

В ATmega328P АЦП реалізовано на основі ЦАП та компаратора. На вхід ЦАП подається зростаючий цифровий код, на виході отиримується зростаюча напруга. Коли напруга з виходу ЦАП зрівняється з вимірюваною напругою з обраного каналу АЦП, процес перетворення можна вважати закінченим та код, що в цей момент був на вході ЦАП, є результатом аналогово-цифрового перетворення. Внаслідок перехідних процесів швидкість АЦП є обмеженою.

### План роботи

1. Навчитися підключати давачі з дискретними вихідними сигналами до Arduino та обробляти дані цих давачів.
2. Навчитися підключати давачі з аналоговими вихідними сигналами до Arduino та обробляти дані цих давачів.

### Порядок виконання роботи

1. Підключити до плати Arduino ультразвуковий давач вимірювання відстані HC-SR04.



Вивід Echo давача підключити до виводу 12 плати Arduino;  
 вивід Trigger – до виводу 11;  
 вивід Vcc – до виводу 5 В;  
 вивід GND – до одного з виводів GND на платі Arduino.

Завантажити в МК наступну програму:

```
#define echoPin 12 // Echo Pin
#define trigPin 11 // Trigger Pin
#define LEDPin 13 // Onboard LED
char rec;
```

```

int maximumRange = 200; // максимальна відстань
int minimumRange = 0; // Minimum range needed
float duration, distance; // Duration used to
calculate distance

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT); // Use LED indicator
  (if required)
}

void loop() {
  /* The following trigPin/echoPin cycle is used
  to determine the distance of the nearest object
  by bouncing soundwaves off of it. */
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);

  //Calculate the distance (in cm) based on the
  speed of sound.
  distance = duration/58.2;

  if (distance >= maximumRange || distance <=
  minimumRange){
    /* Send a negative number to computer and Turn
    LED ON to indicate "out of range" */
    Serial.println("error");
    Serial.print("\n\r");
    digitalWrite(LEDPin, HIGH);
    delay(50);
  }
}

```

```

    }
    else {
        /* Send the distance to the computer using
        Serial protocol, and
        turn LED OFF to indicate successful reading.
        */
        Serial.println(distance);
        Serial.print("\n\r");
        digitalWrite(LEDPin, LOW);
        delay(250);
    }

    //Delay 50ms before next reading.
    delay(150);
    byte l[255];
    byte count=0;
    int i;
    if (Serial.available()>0){
        while(Serial.available()>0){
            digitalWrite(LEDPin, HIGH);
            l[count] = Serial.read();
            count++;}
        for(i=0;i<count;i++){
            Serial.write(l[i]);};
        Serial.println();
        Serial.print("\n\r");
        delay(500);}
}

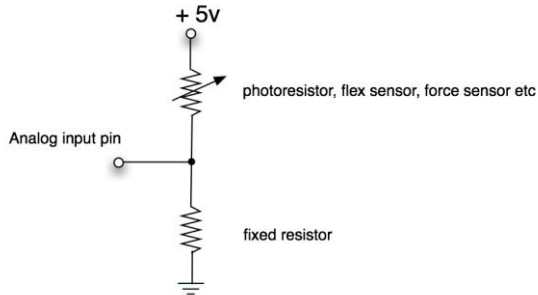
```

Відкрити монітор порту. Піднести руку навпроти давача на відстані 20 – 50 см. Спостерігати за вимірним значенням відстані, що виводиться на екран.

2. Підключити підстроювальний або змінний резистор за схемою подільника напруги (потенціометра).

Відповідно до напруги на ковзному контакті (0...5 В) передавати у послідовний інтерфейс значення змінної `norm`, яка зберігає число від 0 до 100, яке прямопропорційно залежить від напруги.

Variable resistor connected to the analog input of the arduino



```
int sensorPin = A0;    // аналоговий вхід
int sensorValue = 0;   // значення сигналу від
давача
int norm=0; // нормоване значення сигналу з
давача
void setup() {
  Serial.begin(9600);    // налаштуємо
послідовний порт для зв'язку з ПК
}

void loop() {
  sensorValue = analogRead(sensorPin);    //
зчитуємо значення з давача
  norm=map(sensorValue, 0, 1023, 0, 100); //
нормуємо сигнал в межах від 0 до 100
  Serial.println(norm); // виводимо нормоване
значення сигналу в порт
  delay(500); // затримка 0,5 с
}
```

3. Замість змінного резистора підключити до аналогового входу подільник, утворений постійним резистором й давачем освітленості (фоторезистором). Змінюючи освітленість фоторезистора, спостерігати за значенням змінної `norm`.

4. Оформити звіт про виконання лабораторної роботи. Звіт повинен містити: назву та мету лабораторної роботи; тексти програм з коментарями; висновок про виконання роботи.

## Практичне заняття 8

### Тема: Дослідження роботи сервоприводів та реалізація циклограми

**Мета роботи:** Ознайомитися зі способами керування положенням сервоприводів. Навчитися створювати програми керування швидкістю повороту і положенням вихідного вала сервопривода.

#### Теоретичні відомості

Під сервоприводом найчастіше розуміють механізм з електродвигуном, який можна попросити повернутися в заданий кут і утримувати це положення. Однак, це не зовсім повне визначення. Якщо сказати повніше, сервопривід - це привід з управлінням через негативний зворотний зв'язок, що дозволяє точно керувати параметрами руху. Сервоприводом є будь-який тип механічного приводу, що має в складі давач (положення, швидкості, зусилля тощо) і блок керування приводом, який автоматично підтримує необхідні параметри згідно заданому завданню. Тобто сервопривод отримує на вхід значення керуючого параметра, наприклад, кут повороту. Блок управління порівнює це значення зі значенням на своєму датчику.

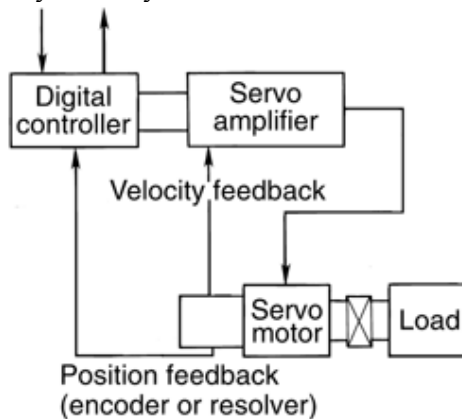


Рис. 8.1. Структурна схема сервопривода з давачем швидкості обертів



Рис. 8.2. Сервоприводи

На основі результату порівняння привод виконує певну дію, наприклад: поворот, прискорення або сповільнення так, щоб значення з внутрішнього датчика стало якомога ближче до значення зовнішнього керуючого параметра. Найбільш поширені сервоприводи, які утримують заданий кут і сервоприводи, що підтримують задану швидкість обертання.

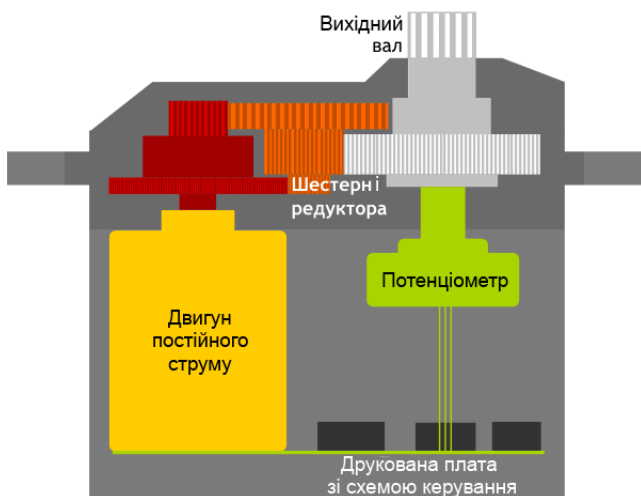


Рис. 8.3. Будова сервопривода

Отже, сервопривод – це регульований редукторний електродвигун. Він зазвичай складається з приводного механізму з двигуном постійного струму, плати управління і потенціометра, котрий забезпечує зворотний зв'язок.

### Управління сервоприводом. Інтерфейс керуючих сигналів

Керуючий сигнал для сервопривода – імпульси постійної частоти і змінної ширини. Те, яке положення повинен зайняти сервопривод, залежить від довжини імпульсів. Коли сигнал надходить в керуючу схему, наявний у ній генератор імпульсів виробляє свій імпульс, тривалість якого визначається через потенціометр. Інша частина схеми порівнює тривалість двох імпульсів. Якщо тривалість різна, включається електродвигун. Напрямок обертання визначається тим, який з імпульсів коротший. Якщо довжини імпульсів рівні, електродвигун зупиняється. Найчастіше в аматорських сервоприводах імпульси виробляються з частотою 50 Гц. Це означає, що імпульс випускається і приймається раз в 20 мс. Зазвичай при цьому тривалість імпульсу 1520 мкс означає, що сервопривод повинен зайняти середнє положення. Збільшення або зменшення довжини імпульсу змусить сервопривод повернутися за годинниковою або проти годинникової стрілки відповідно. При цьому існують верхня і нижня межі тривалості імпульсу.

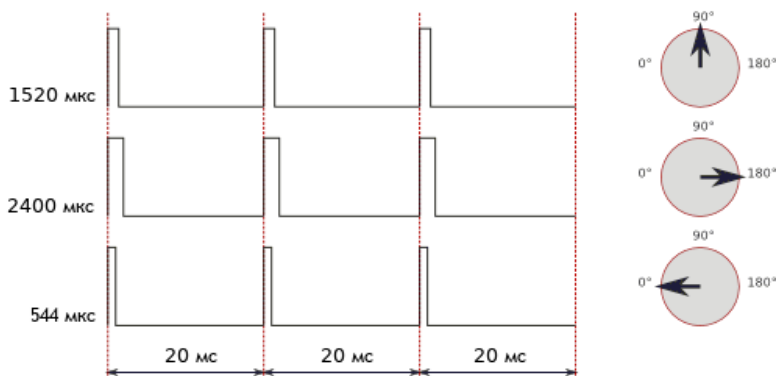


Рис. 8.4. Залежність кута повороту сервопривода від тривалості керуючих імпульсів

У бібліотеці Servo для Arduino за замовчуванням виставлені наступні значення довжин імпульсу: 544 мкс – для 0° і 2400 мкс –

для 180°. На конкретному пристрої заводські налаштування можуть відрізнятися від стандартних. Також варто відзначити, що це всього лише загальноприйняті довжини. Навіть у рамках однієї і тієї ж моделі сервоприводу може існувати похибка, що допускається при виробництві, яка призводить до того, що робочий діапазон довжин імпульсів трохи відрізняється. Для точної роботи кожен конкретний сервопривод повинен бути відкалібрований: шляхом експериментів необхідно підібрати коректний діапазон, характерний саме для нього. При формуванні сигналу керування для сервопривода важливою є довжина імпульсів, а не частота їх появи. 50 Гц – це норма, але сервопривод буде працювати коректно і при 40, і при 60 Гц. При цьому слід мати на увазі, що при сильному зменшенні частоти він може працювати ривками і на зниженій потужності, а при сильному завищенні частоти може перегрітися і вийти з ладу.

Сервоприводи малої потужності (наприклад, SG90) можна безпосередньо підключати до плати Arduino, для підключення більш потужних – слід використовувати зовнішнє джерело живлення.

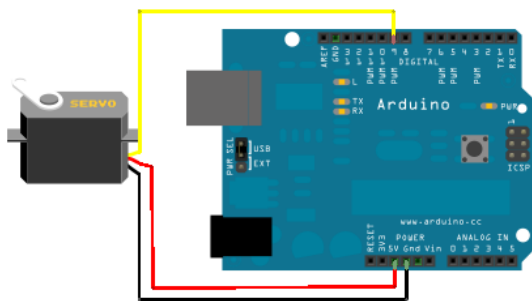


Рис. 8.5. Підключення сервопривода малої потужності до мікроконтролерної плати Arduino Uno

Бібліотека Servo дозволяє здійснювати програмне керування сервоприводами. Управління здійснюється наступними функціями:

`attach()` – приєднує змінну до конкретного виводу плати. Можливі два варіанти синтаксису для цієї функції: `servo.attach(pin)` і `servo.attach (pin, min, max)`. При цьому `pin` - номер виводу, до якого приєднують сервопривод, `min` і `max` - довжини імпульсів в



мікросекундах, що відповідають за кути повороту 0° і 180°. За замовчуванням виставляються рівними 544 мкс і 2400 мкс відповідно.

`write()` - віддає команду сервоприводу прийняти деяке значення параметра. Синтаксис наступний:

`servo.write (angle)`, де `angle` - кут, на який повинен повернутися сервопривод.

`writeMicroseconds()` - віддає команду надіслати на сервопривід імпульс певної довжини, є низькорівневим аналогом попередньої команди. Синтаксис наступний:

`servo.writeMicroseconds(uS)`, де `uS` – довжина імпульсу в мікросекундах.

`read()` - читає поточне значення кута, в якому знаходиться сервопривод. Синтаксис наступний: `servo.read()`, повертається ціле значення від 0 до 180.

`attached()` - перевірка, чи була приєднана змінна до конкретного виводу. Синтаксис наступний: `servo.attached()`, повертається `true`, якщо змінна була приєднана до якого-небудь виводу, або `false` в зворотному випадку.

`detach()` - виконує дію, зворотну дії `attach()`, тобто від'єднує змінну від виводу, до якого вона була приписана. Синтаксис наступний: `servo.detach()`.

### ***Приклад програми з використанням бібліотеки Servo***

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(9);
}
void loop()
{
    myservo.write(90); // встановлюємо сервопривод в
    середнє положення
    delay(500);
    myservo.write(0); // встановлюємо сервопривод
    в крайнє ліве положення
    delay(500);
    myservo.write(180); // встановлюємо
```

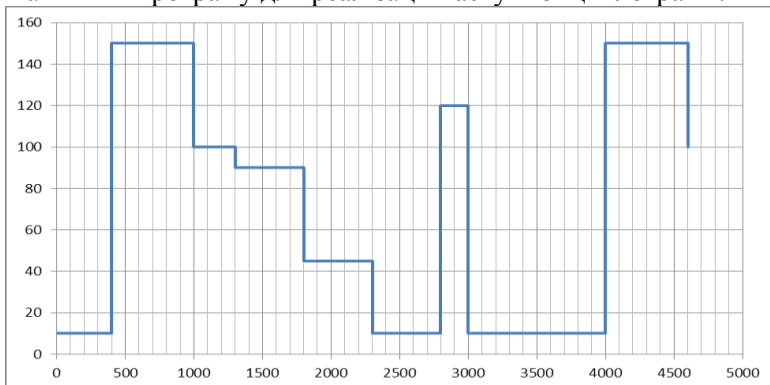
```
сервопривод в крайнє праве положення
    delay(500);
}
```

### План роботи

1. Скласти програму для керування положенням сервопривода.
2. Реалізувати циклограму положення вала сервопривода.
3. Програмно обмежити швидкість повороту вала сервопривода.

### Порядок виконання роботи

1. Складіть схему за рис. 8.5, використовуючи сервопривод SG90.
  2. Завантажте програму з прикладу бібліотеки Servo.
- Напишіть програму для реалізації наступної циклограми.



Вісь абсцис – час в мс, вісь ординат – положення вала сервопривода в градусах.

3. Для того, щоб мати змогу змінювати швидкість обертання використайте бібліотеку VarSpeedServo. Завантажити в МК приклад VarSpeedServo, змінюючи значення швидкості. Переконайтеся в правильності роботи програми та точності позиціювання сервопривода.

4. Оформити звіт про виконання лабораторної роботи. Звіт повинен містити: назву та мету лабораторної роботи; тексти програм з коментарями; висновок про виконання роботи.

## Рекомендована література

### Базова

1. Синтез робототехнічних систем в машинобудуванні / Л. Є. Пелевін, К. І. Почка, О. М. Гаркавенко та ін. К. : Інтерсервіс, 2016. 258 с.
2. 1. Ніколайчук В. М. Основи робототехніки : навч. посіб. Рівне : НУВГП, 2008. 76 с. URL : <http://ep3.nuwm.edu.ua/2243/>.
3. Робототехніка / В. І. Костюк, Г. О. Спишу, Л. С. Ямпольський та ін. К. : Вища школа, 1994. 447 с.
4. Проць Я. І. Захоплювальні пристрої промислових роботів: Навчальний посібник. Тернопіль : Тернопільський державний технічний університет ім. І. Пулюя, 2008. 232 с.
5. Головкин Д. Б., Рогов К. Г., Скрипник Ю. О. Автоматика і автоматизація технологічних процесів. К. : Либідь, 1997. 326 с.

### Допоміжна

1. Пелевін Л. Є., Балака М. М., Аржаєв Г. О. Механотронні системи гідро пневмоавтоматики. К. : Аграр Медіа Груп, 2014. 192 с.
2. Юревич Е. И. Основы робототехники: учебное пособие для вузов / 2-е изд. Санкт-Петербург: БХВ-Петербург, 2007. 401 с.
3. Попок Т. В. Штучний інтелект: перспективи та загрози. *Студентський вісник НУВГП*. Рівне : НУВГП, 2015. Вип. 2(4). С. 252-253. URL : <http://ep3.nuwm.edu.ua/4149/>.
4. Спишу Г. А. Промышленные роботы: Конструирование и применение. К. : Вища школа, 1991. 311 с.
5. Павленко И. И. Структура промислових роботів. Кіровоград : РВЛ, 1998. 98 с.